

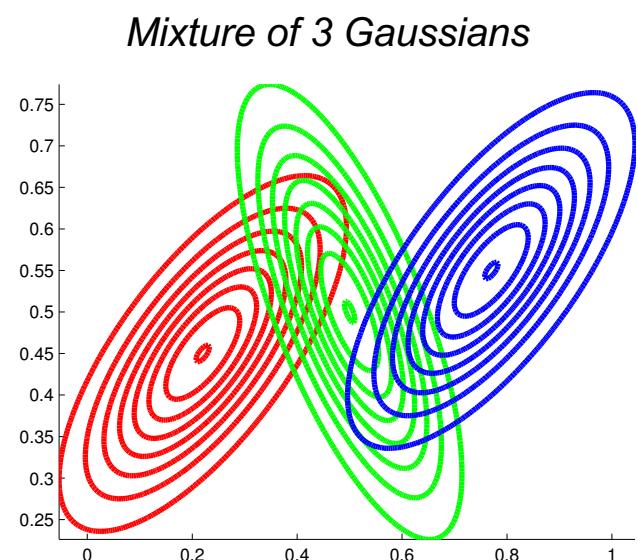
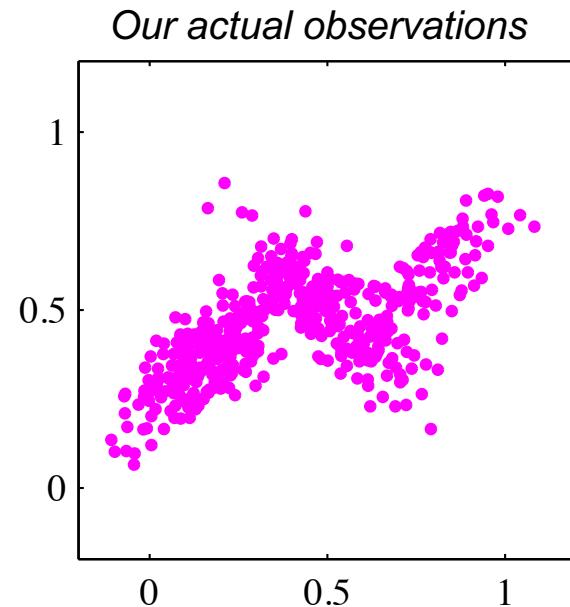
Original em Slides

Expectation Maximization for Mixtures of Gaussians

CS229: Machine Learning
Carlos Guestrin
Stanford University

©2022 Carlos Guestrin

Learning a Mixture of Gaussians



Summary of GMM Components

- Observations $x^i \in \mathbb{R}^d, i = 1, 2, \dots, N$
- Hidden cluster labels $z_i \in \{1, 2, \dots, K\}, i = 1, 2, \dots, N$
- Hidden mixture means $\mu_k \in \mathbb{R}^d, k = 1, 2, \dots, K$
- Hidden mixture covariances $\Sigma_k \in \mathbb{R}^{d \times d}, k = 1, 2, \dots, K$
- Hidden mixture probabilities $\pi_k, \sum_{k=1}^K \pi_k = 1$

Gaussian mixture marginal and conditional likelihood :

$$p(x^i | \pi, \mu, \Sigma) = \sum_{z^i=1}^K \pi_{z^i} p(x^i | z^i, \mu, \Sigma)$$

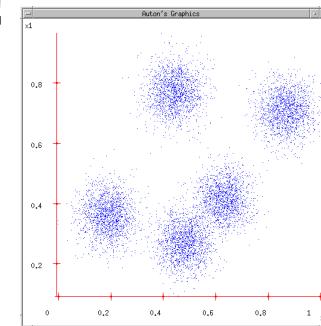
$$p(x^i | z^i, \mu, \Sigma) = \mathcal{N}(x^i | \mu_{z^i}, \Sigma_{z^i})$$

©2022 Carlos Guestrin

CS229: Machine Learning

But we don't see class labels!!!

- MLE:
 - $\operatorname{argmax} \prod_i P(z^i, x^i)$
- But we don't know z^i
- Maximize marginal likelihood:
 - $\operatorname{argmax} \prod_i P(x^i) = \operatorname{argmax} \prod_i \sum_k P(z^i=k, x^i)$



Special case: spherical Gaussians and hard assignments

$$P(z^i = k, \mathbf{x}^i) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x}^i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}^i - \boldsymbol{\mu}_k)\right] P(z^i = k)$$

- If $P(\mathbf{X}|z=k)$ is spherical, with same σ for all classes:

$$P(\mathbf{x}^i | z^i = k) \propto \exp\left[-\frac{1}{2\sigma^2} \|\mathbf{x}^i - \boldsymbol{\mu}_k\|^2\right]$$

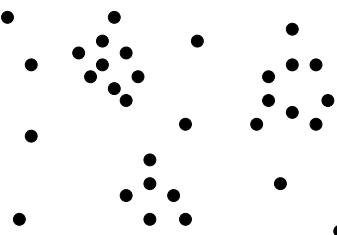
- If each x^i belongs to one class $C(i)$ (hard assignment), marginal likelihood:

$$\prod_{i=1}^N \sum_{k=1}^K P(\mathbf{x}^i, z^i = k) \propto \prod_{i=1}^N \exp\left[-\frac{1}{2\sigma^2} \|\mathbf{x}^i - \boldsymbol{\mu}_{C(i)}\|^2\right]$$

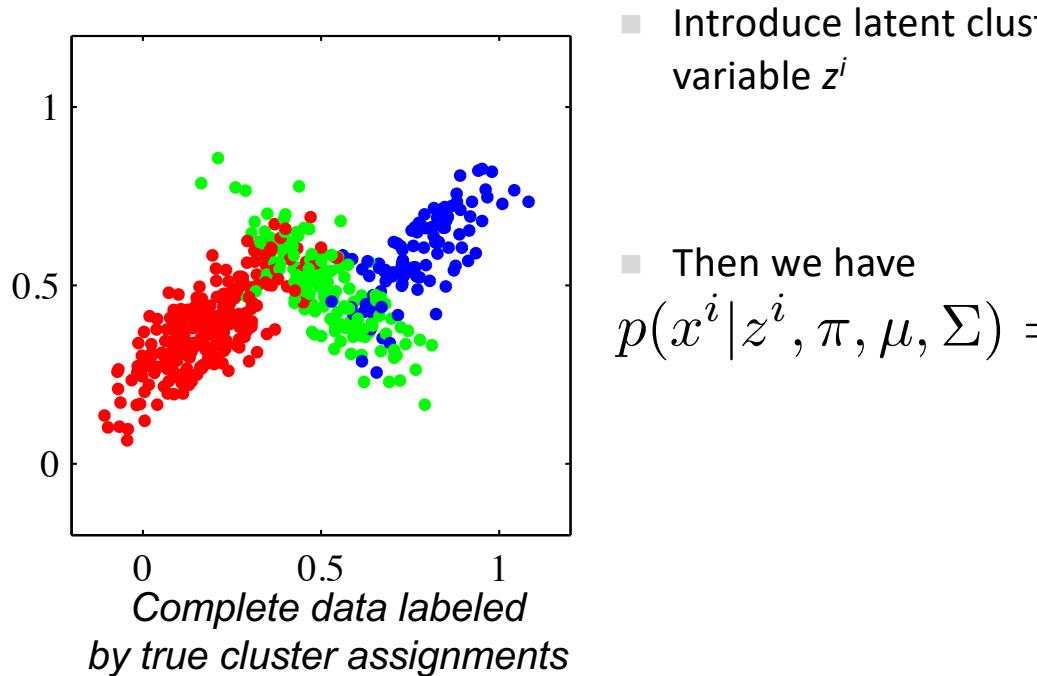
- Same as K-means!!!

EM: “Reducing” Unsupervised Learning to Supervised Learning

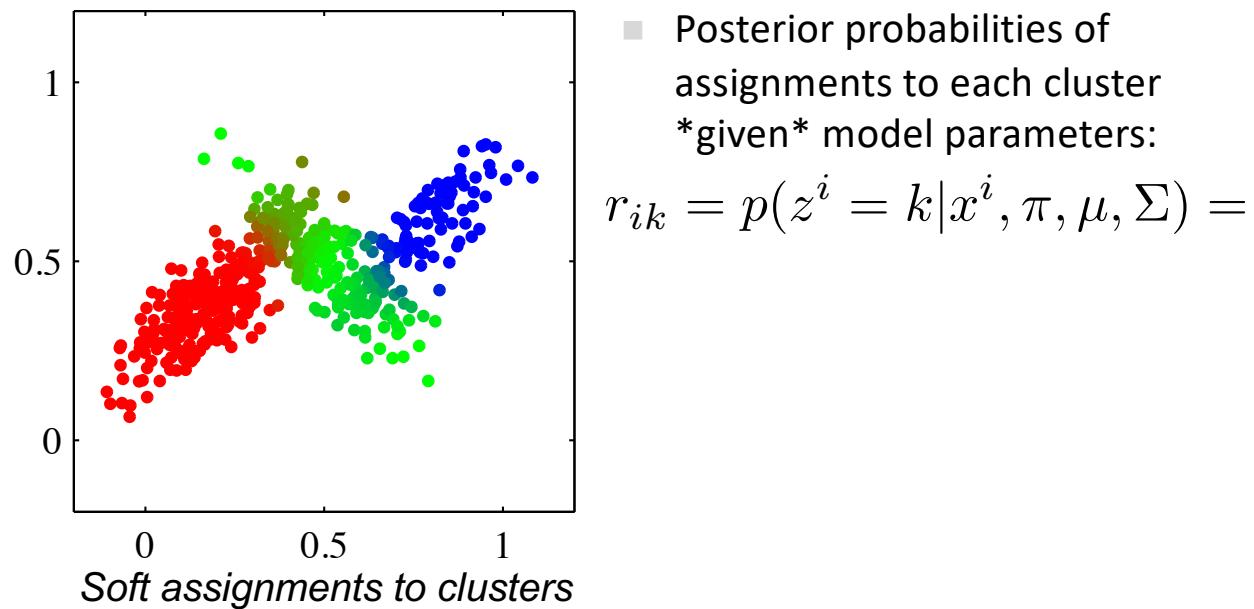
- If we knew assignment of points to classes → Supervised Learning!
- Expectation-Maximization (EM)
 - **Expectation:** Guess assignment of points to classes
 - In standard (“soft”) EM: each point associated with prob. of being in each class
 - **Maximization:** Recompute model parameters
 - Iterate



Imagine we have an assignment of each x^i to a Gaussian



Expectation: infer cluster assignments from observations



ML Estimate of Mixture Model Params

- Log likelihood

$$L_x(\theta) \triangleq \log p(\{x^i\} \mid \theta) = \sum_{i=1}^N \log \sum_{j=1}^K p(x^i, z=j \mid \theta)$$

- Want ML estimate

$$\hat{\theta}^{ML} =$$

- Neither convex nor concave and local optima

Maximization: If “complete” data were observed...

- Assume class labels z^i were observed in addition to x^i

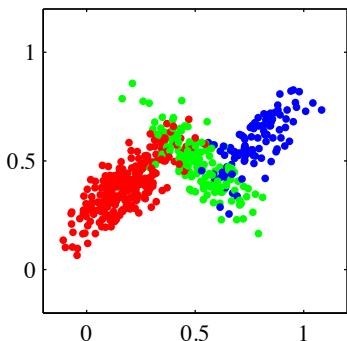
$$L_{x,z}(\theta) = \sum_{i=1}^N \log p(x^i, z^i | \theta)$$

- Compute ML estimates

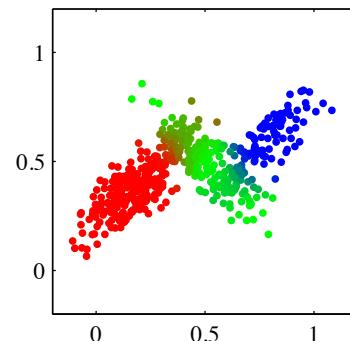
- Separates over clusters k !

- Example: mixture of Gaussians (MoG) $\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$

Maximization: if inferred cluster assignments from observations



Complete data labeled
by true cluster assignments



Soft assignments to clusters

- Posterior probabilities of assignments to each cluster *given* model parameters:

$$r_{ik} = p(z^i = k | x^i, \pi, \mu, \Sigma)$$

Expectation-Maximization Algorithm

- Motivates a coordinate ascent-like algorithm:
 1. Infer missing values z^i given estimate of parameters $\hat{\theta}$
 2. Optimize parameters to produce new $\hat{\theta}$ given “filled in” data z^i
 3. Repeat

- Example: MoG

1. Infer “responsibilities”

$$r_{ik} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)})$$

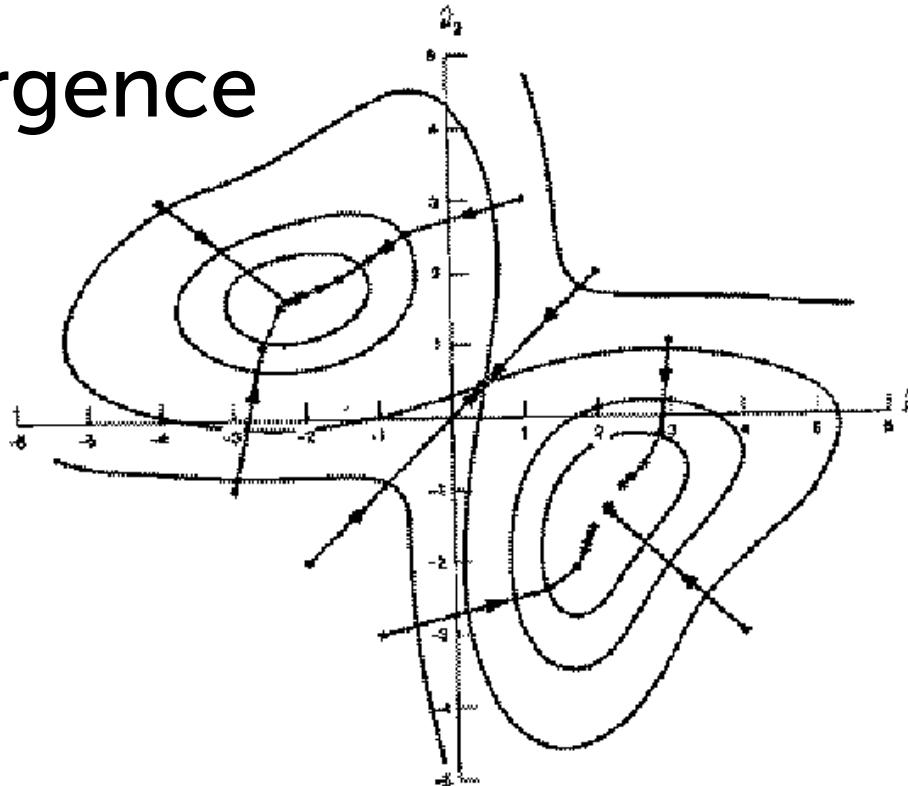
2. Optimize parameters

max w.r.t. π_k :

max w.r.t. μ_k, Σ_k :

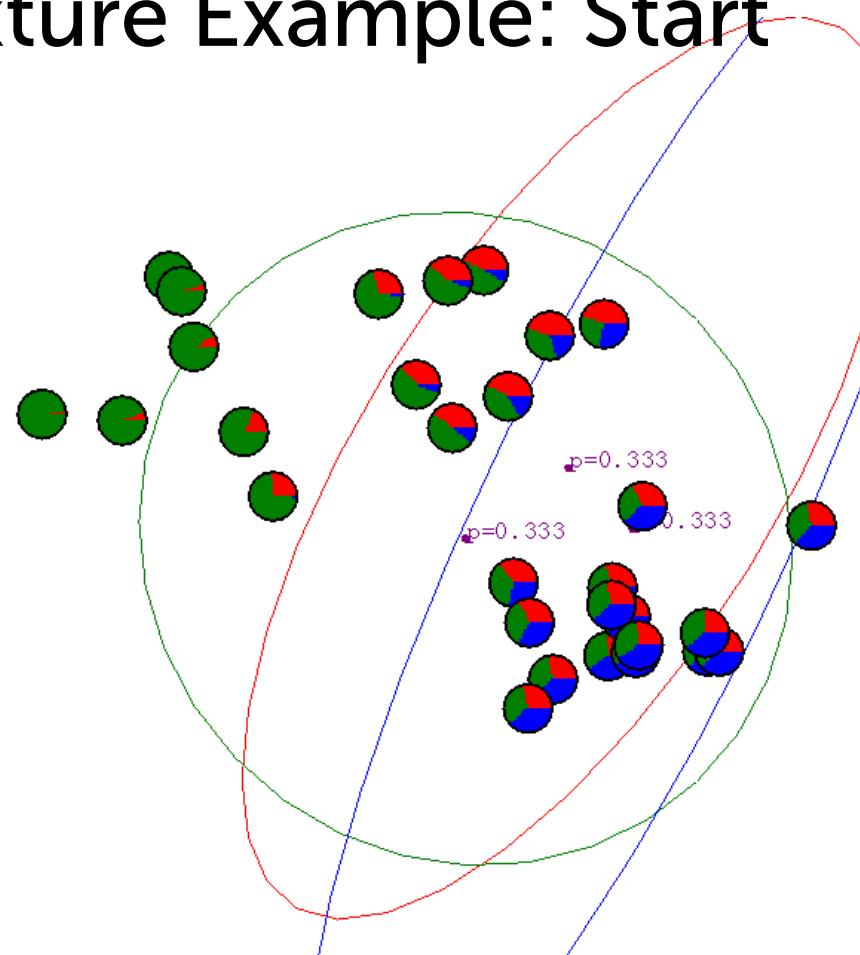
E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. → convergence to a local optimum guaranteed

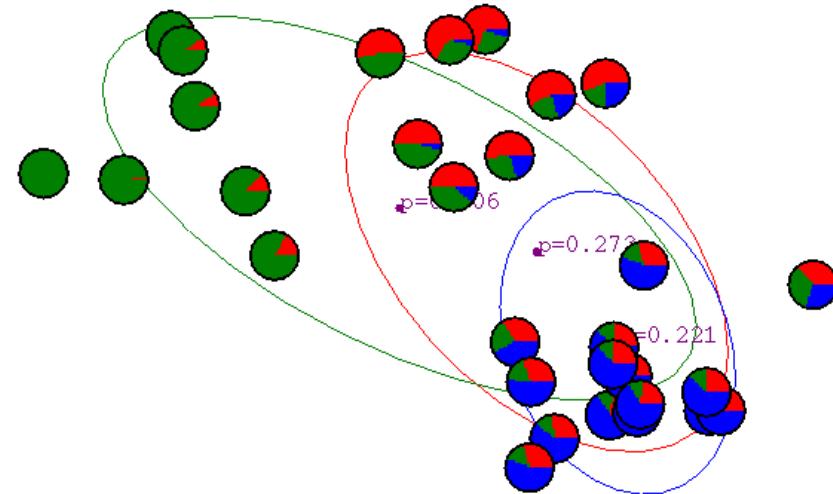


- This algorithm is REALLY USED. And in high dimensional state spaces, too.

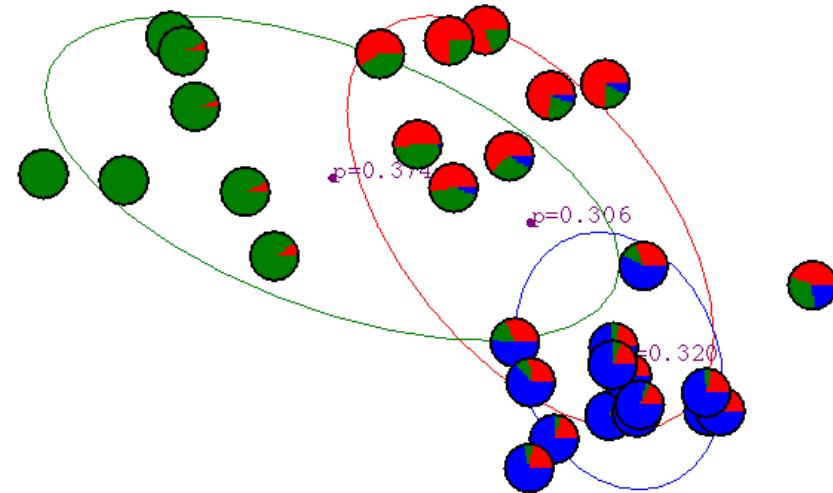
Gaussian Mixture Example: Start



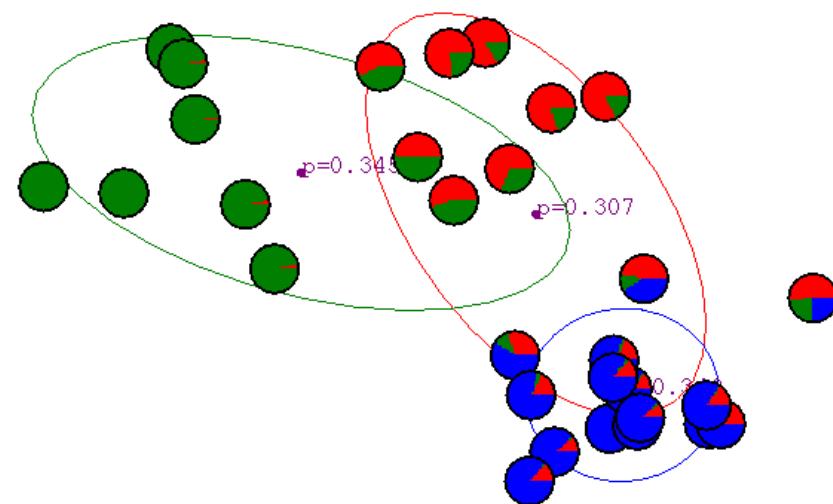
After first iteration



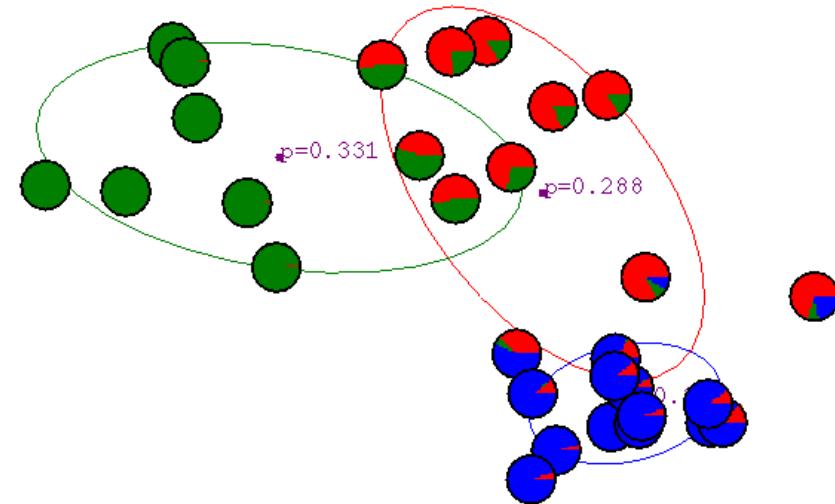
After 2nd iteration



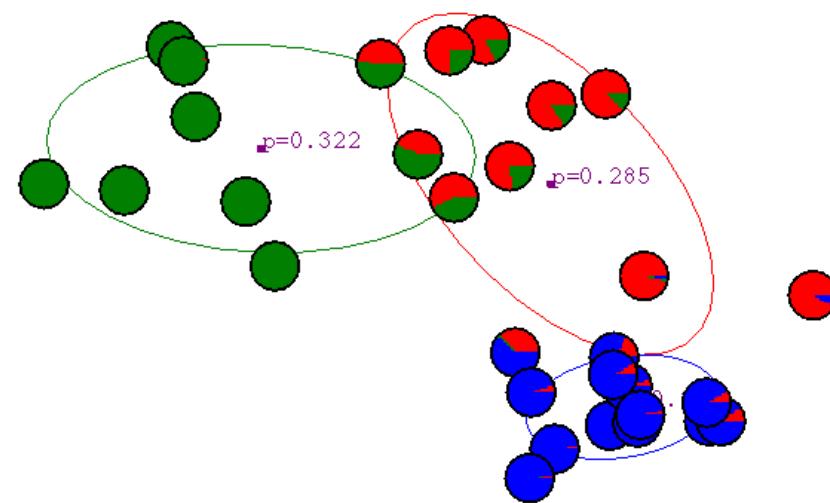
After 3rd iteration



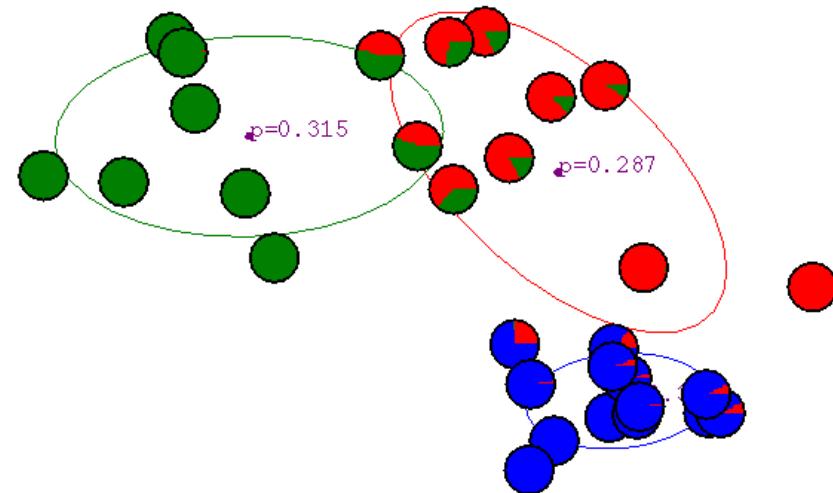
After 4th iteration



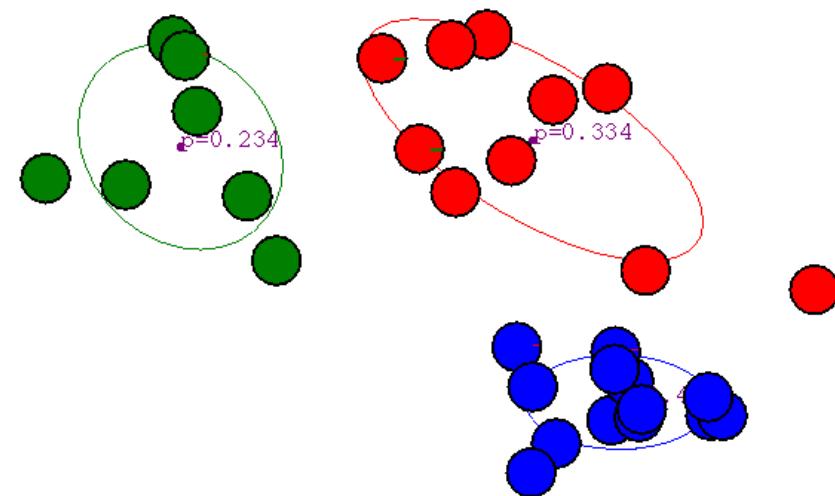
After 5th iteration



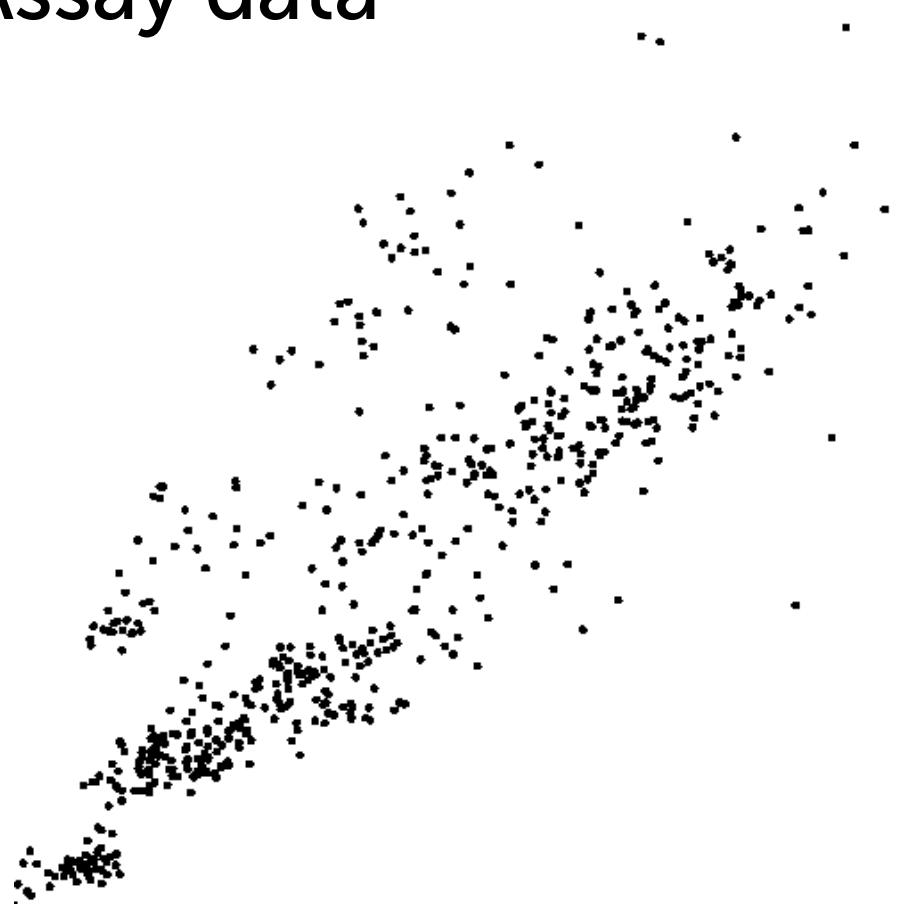
After 6th iteration



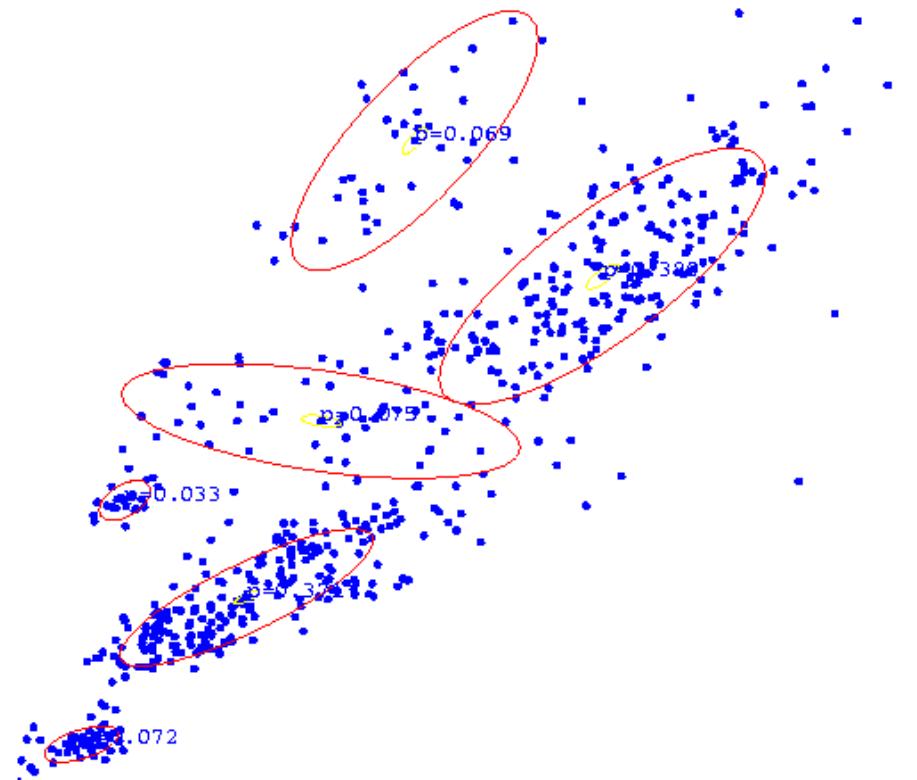
After 20th iteration



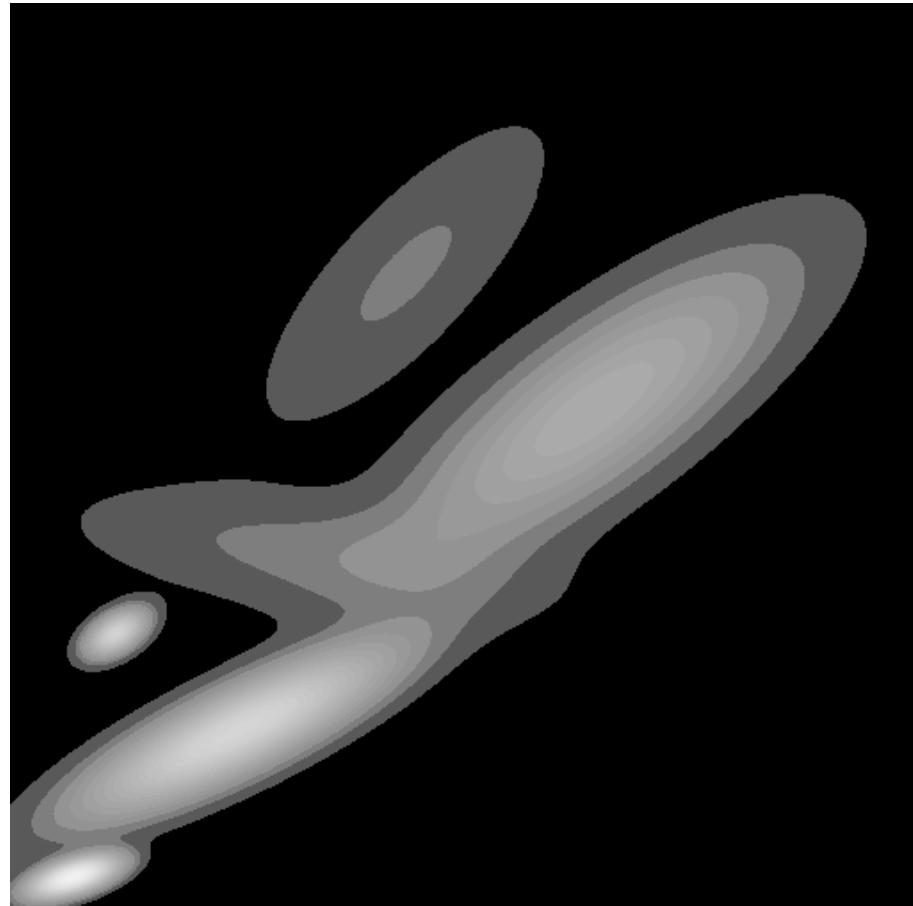
Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator



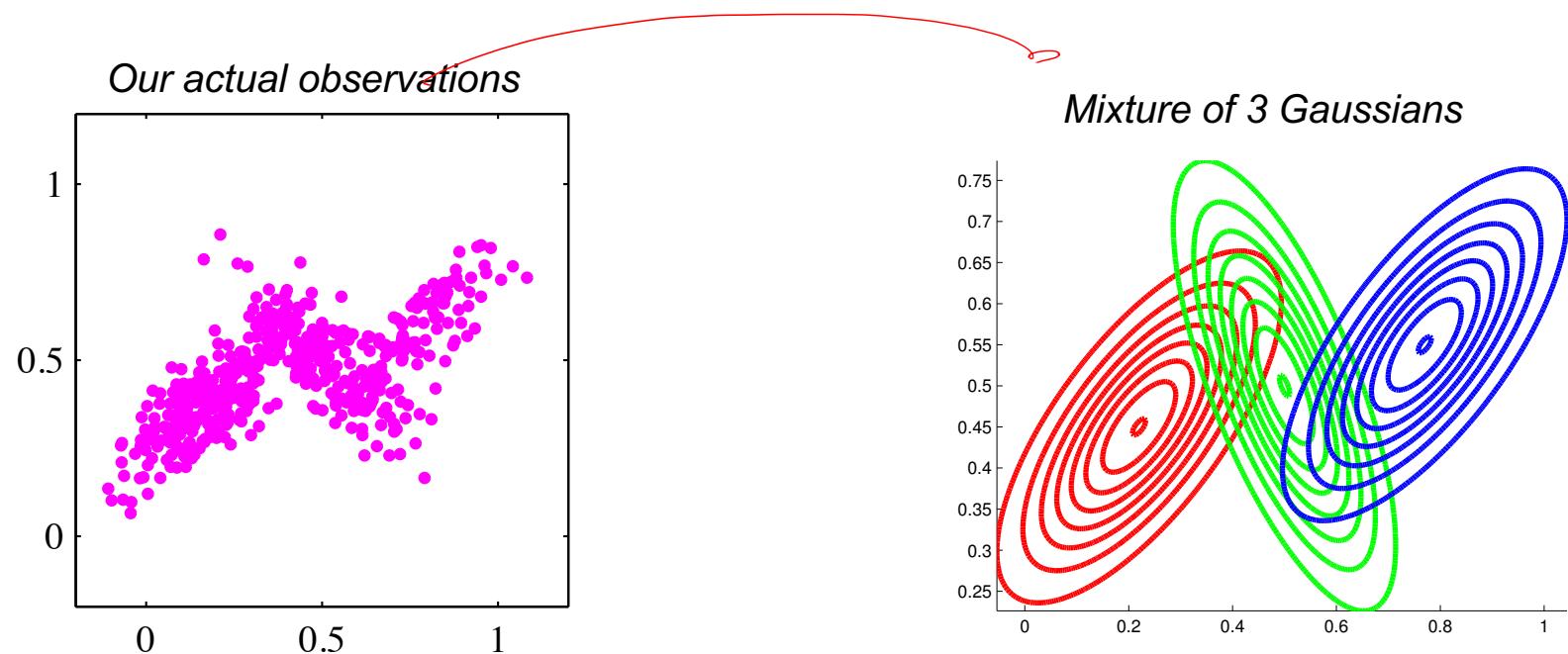
Annotated em Slides

Expectation Maximization for Mixtures of Gaussians

CS229: Machine Learning
Carlos Guestrin
Stanford University

©2022 Carlos Guestrin

Learning a Mixture of Gaussians



Summary of GMM Components

- Observations $x^i \in \mathbb{R}^d, i = 1, 2, \dots, N$
- Hidden cluster labels $z_i \in \{1, 2, \dots, K\}, i = 1, 2, \dots, N$
- Hidden mixture means $\mu_k \in \mathbb{R}^d, k = 1, 2, \dots, K$
- Hidden mixture covariances $\Sigma_k \in \mathbb{R}^{d \times d}, k = 1, 2, \dots, K$
- Hidden mixture probabilities $\pi_k, \sum_{k=1}^K \pi_k = 1$

Gaussian mixture marginal and conditional likelihood :

$$p(x^i | \pi, \mu, \Sigma) = \sum_{z^i=1}^K \pi_{z^i} p(x^i | z^i, \mu, \Sigma)$$

$$p(x^i | z^i, \mu, \Sigma) = \mathcal{N}(x^i | \mu_{z^i}, \Sigma_{z^i})$$

But we don't see class labels!!!

- MLE:

- $\operatorname{argmax} \prod_i P(z^i, x^i)$

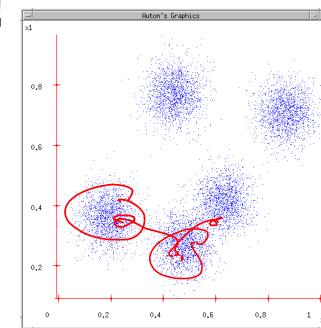
latent variable

- But we don't know z^i
- Maximize marginal likelihood:

- $\operatorname{argmax} \prod_i P(x^i) = \operatorname{argmax} \prod_i \sum_k P(z^i=k, x^i)$

μ, Σ, π

*part of f_{θ}
Observed data*



$$\log \pi = \sum \log$$

Special case: spherical Gaussians and hard assignments

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

$$P(z^i = k, \mathbf{x}^i) = \frac{1}{(2\pi)^{m/2} |\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x}^i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}^i - \mu_k)\right] P(z^i = k)$$

$N(\mu_k, \Sigma_k)$

cluster probability

- If $P(X|z=k)$ is spherical, with same σ for all classes: $P(z^i = k) = P(z^i = j)$

$$P(\mathbf{x}^i | z^i = k) \propto \exp\left[-\frac{1}{2\sigma^2} \|\mathbf{x}^i - \mu_k\|^2\right]$$

K-means objective

- If each x^i belongs to one class $C(i)$ (hard assignment), marginal likelihood:

$$\underset{\text{max}}{\text{Want to}} \quad \prod_{i=1}^N \sum_{k=1}^K P(\mathbf{x}^i, z^i = k) \propto \prod_{i=1}^N \exp\left[-\frac{1}{2\sigma^2} \|\mathbf{x}^i - \mu_{C(i)}\|^2\right]$$

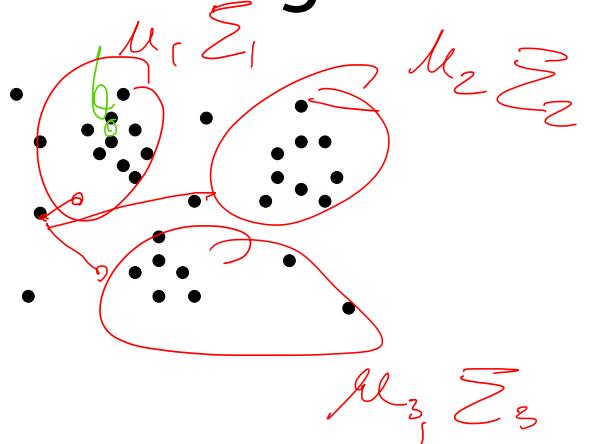
$$\min_{\boldsymbol{\mu}} \sum_{i=1}^N \|\mathbf{x}^i - \mu_{C(i)}\|_2^2$$

- Same as K-means!!!

$$\max_i \sum_{i=1}^N \frac{-1}{2\sigma^2} \|\mathbf{x}^i - \mu_{C(i)}\|_2^2$$

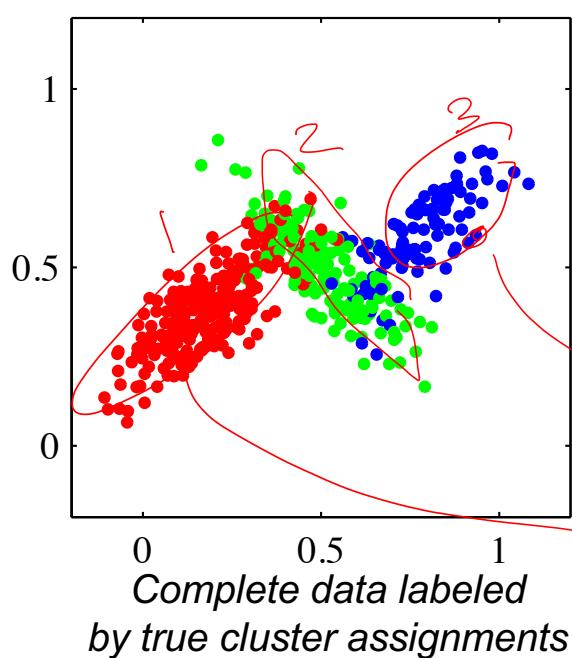
EM: “Reducing” Unsupervised Learning to Supervised Learning

- If we knew assignment of points to classes → Supervised Learning!
- Expectation-Maximization (EM)
 - Expectation: Guess assignment of points to classes
 - In standard (“soft”) EM: each point associated with prob. of being in each class
 - Maximization: Recompute model parameters
 - Iterate



as if we had
fully observed data

Imagine we have an assignment of each x^i to a Gaussian

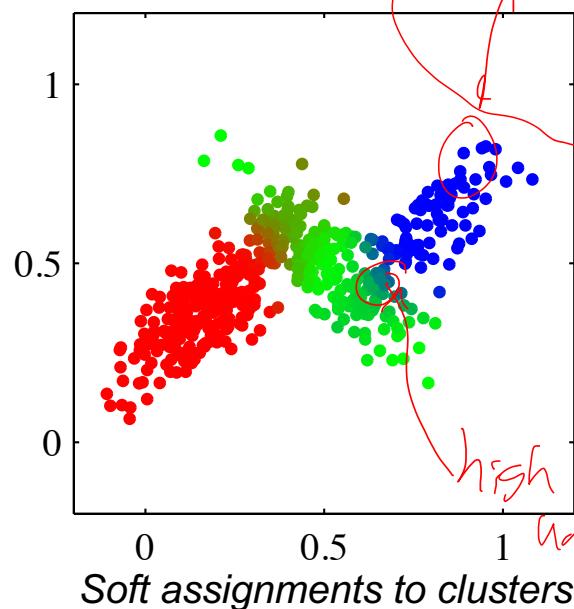


- Introduce latent cluster indicator variable z^i
 $z^i \in \{1, \dots, k\}$
 $P(z^i = k) = \pi_k$
- Then we have
 $p(x^i | z^i, \pi, \mu, \Sigma) = N(x^i | \mu_{z^i}, \Sigma_{z^i})$
 $N(\mu_1, \Sigma_1)$
 $N(\mu_2, \Sigma_2)$
 $N(\mu_3, \Sigma_3)$

Expectation: infer cluster assignments from observations

Normalization:

$$\sum_{k=1}^K r_{ik} = 1$$



- Posterior probabilities of assignments to each cluster *given* model parameters: π, μ, Σ

$$\begin{aligned} r_{ik} &= p(z^i = k | x^i, \pi, \mu, \Sigma) = \\ &= \frac{P(z^i = k) P(x^i | z^i = k)}{P(x^i)} \quad \text{normalize} \\ &= \frac{\pi_k N(x^i | \mu_k, \Sigma_k)}{P(x^i)} \end{aligned}$$

ML Estimate of Mixture Model Params

(π, μ, Σ)

- Log likelihood

$$L_x(\theta) \triangleq \log p(\{x^i\} \mid \theta) = \sum_{i=1}^N \log \sum_{j=1}^K p(x^i, z=j \mid \theta)$$

- Want ML estimate

$$\hat{\theta}^{ML} = \underset{\theta}{\operatorname{arg\,max}} \ L_x(\theta)$$

- Neither convex nor concave and local optima

$$\theta = \{\pi, \mu, \Sigma\}$$

Maximization: If "complete" data were observed...

Solve as two learning problems

- Assume class labels z^i were observed in addition to x^i

MLE:

$$L_{x,z}(\theta) = \sum_{i=1}^N \log p(x^i, z^i | \theta) = \sum_{i=1}^N \log P(z^i | \pi) + \sum_{i=1}^N \log P(x^i | z^i, \mu, \Sigma)$$

$$\hat{\pi}_j = \frac{\text{Count}(z^i=j)}{N} = \sum_{i=1}^N \log P(z^i | \pi)$$

- Compute ML estimates

 - Separates over clusters $k!$

*K indep.
learning problems*

- Example: mixture of Gaussians (MoG)

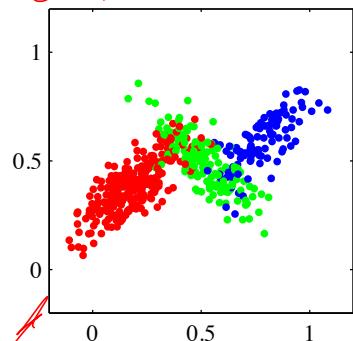
$$\sum_{j=1}^K \sum_{i: z^i=j} \log P(x^i | \mu_j, \Sigma_j)$$

$$\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$$

MLE $\hat{\mu}_j$ = mean of x^i assigned to cluster j

Maximization: if inferred cluster assignments from observations

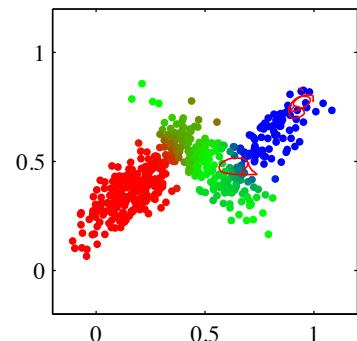
Fully observed data



Complete data labeled by true cluster assignments

estimate π, μ, Σ
using simple
Counts & averages

e.g.



Soft assignments to clusters

- Posterior probabilities of assignments to each cluster *given* model parameters:

$$r_{ik} = p(z^i = k | x^i, \pi, \mu, \Sigma)$$

Weighted counts()

$$\hat{\pi}_j = \frac{N}{\sum_i r_{ij}}$$

$$\hat{\mu}_j = \frac{\sum_i r_{ij} x^i}{N}$$

$$\hat{\Sigma}_j = \frac{\sum_i r_{ij} (x^i - \hat{\mu}_j)(x^i - \hat{\mu}_j)^T}{N}$$

Expectation-Maximization Algorithm

- Motivates a coordinate ascent-like algorithm:

E 1. Infer missing values z^i given estimate of parameters $\hat{\theta}$

M 2. Optimize parameters to produce new $\hat{\theta}$ given “filled in” data z^i
3. Repeat

- Example: MoG

- Infer “responsibilities”

$$r_{ik} = p(z^i = k \mid x^i, \hat{\theta}^{(t-1)}) \leftarrow \text{Bayes rule}$$

- Optimize parameters

max w.r.t. π_k :

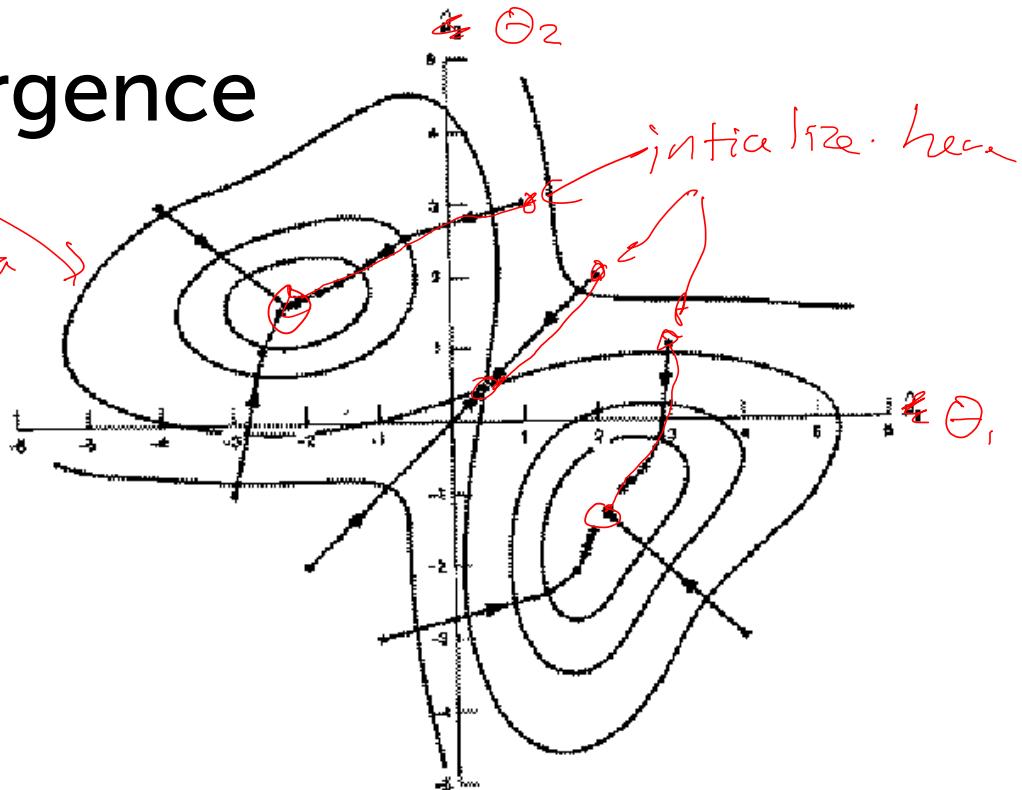
MLE on weighted data

max w.r.t. μ_k, Σ_k :

E.M. Convergence

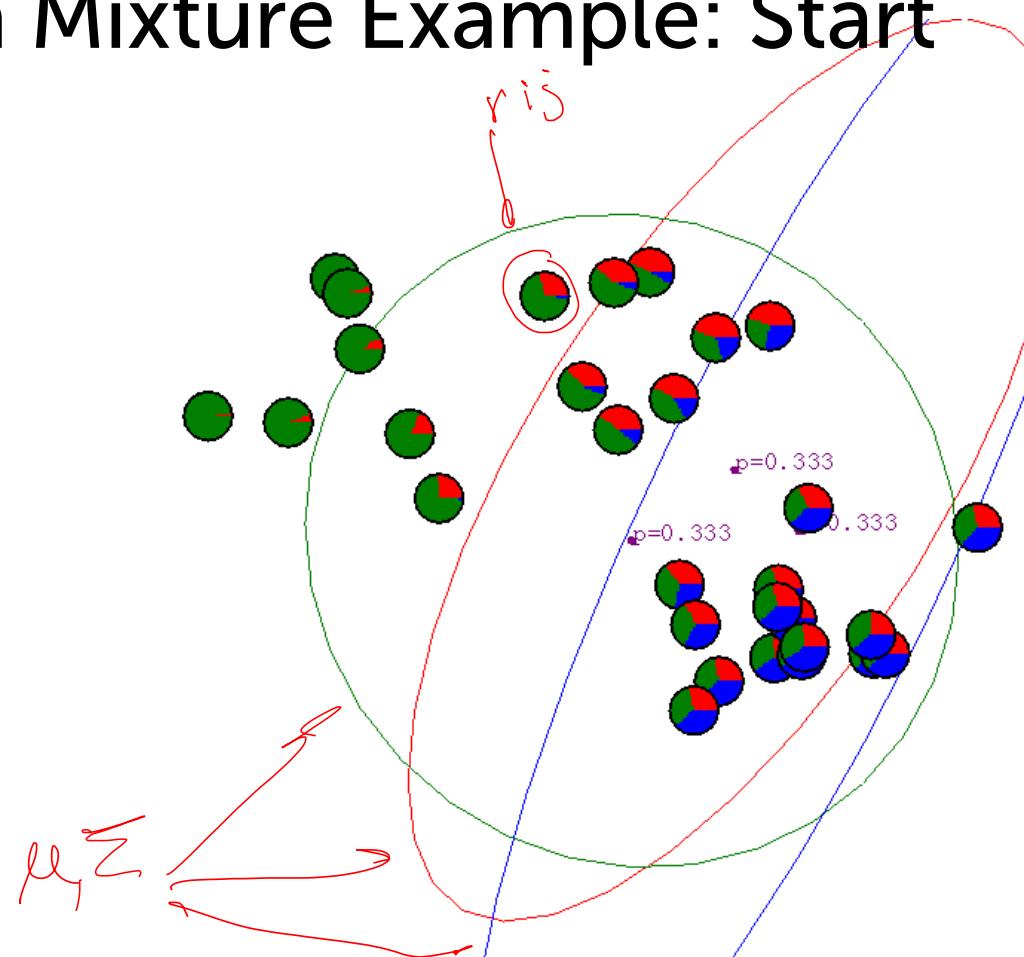
level sets
of loss function
on unlabeled data

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. \rightarrow convergence to a local optimum guaranteed

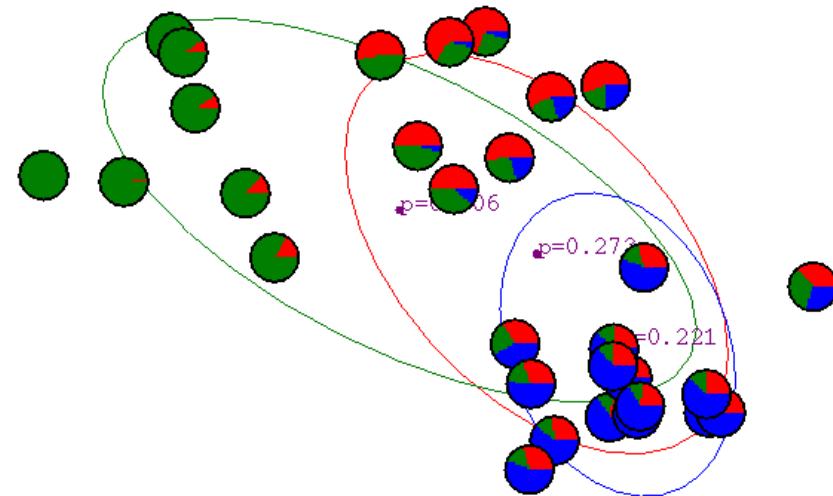


- This algorithm is REALLY USED. And in high dimensional state spaces, too.

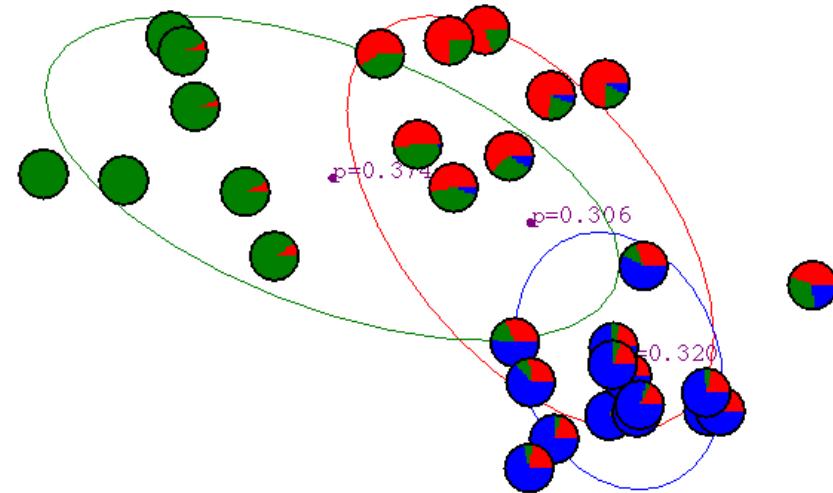
Gaussian Mixture Example: Start



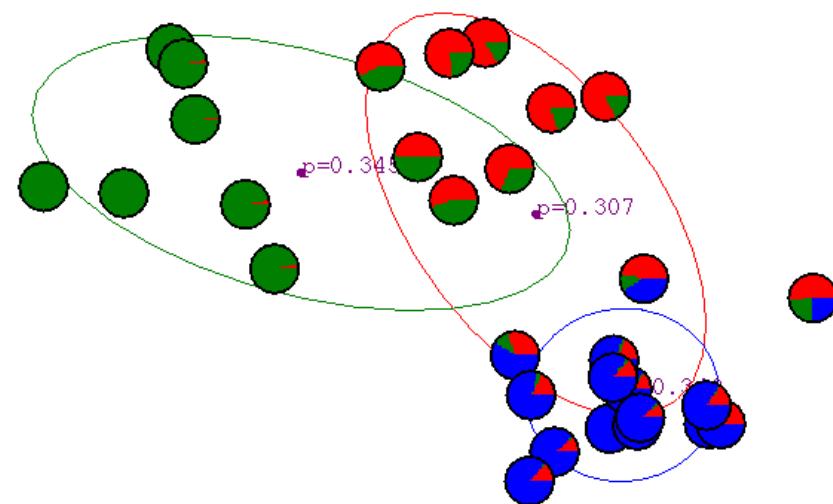
After first iteration



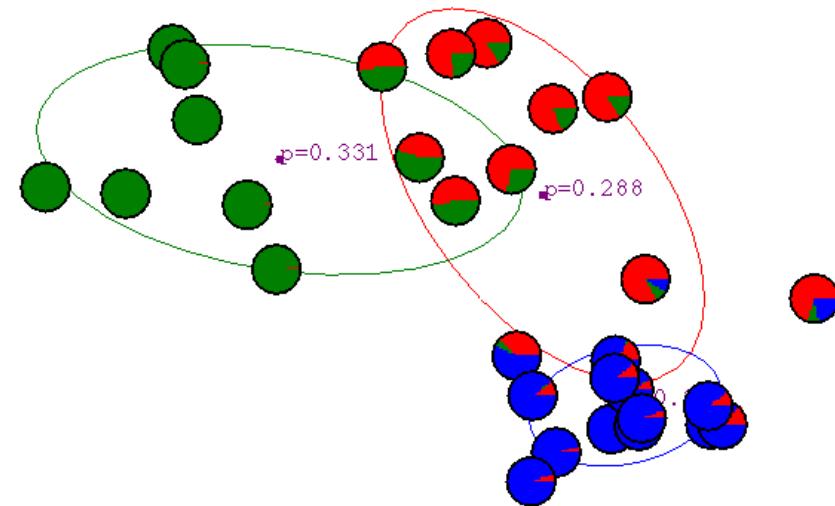
After 2nd iteration



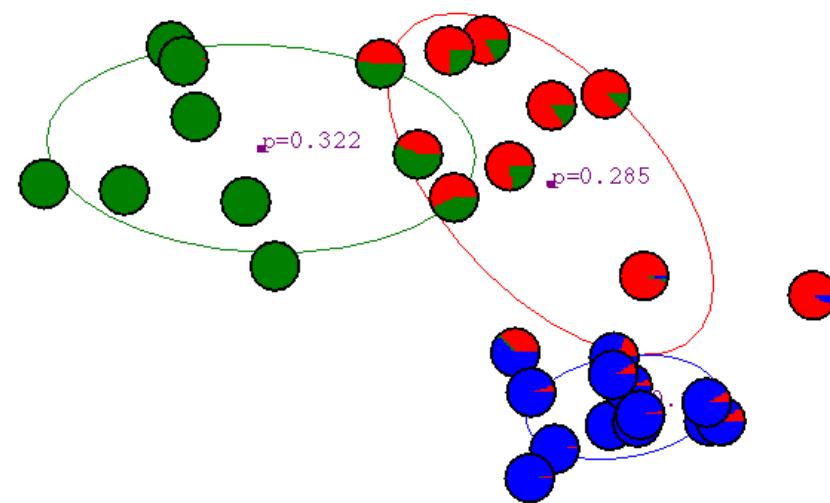
After 3rd iteration



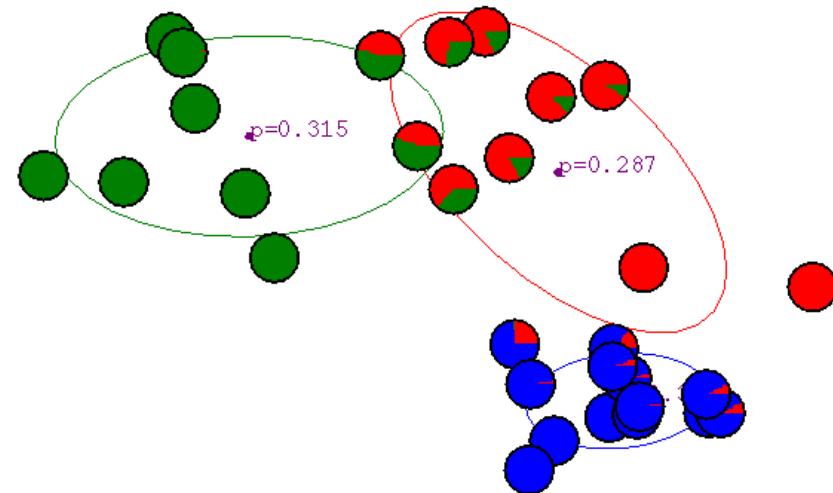
After 4th iteration



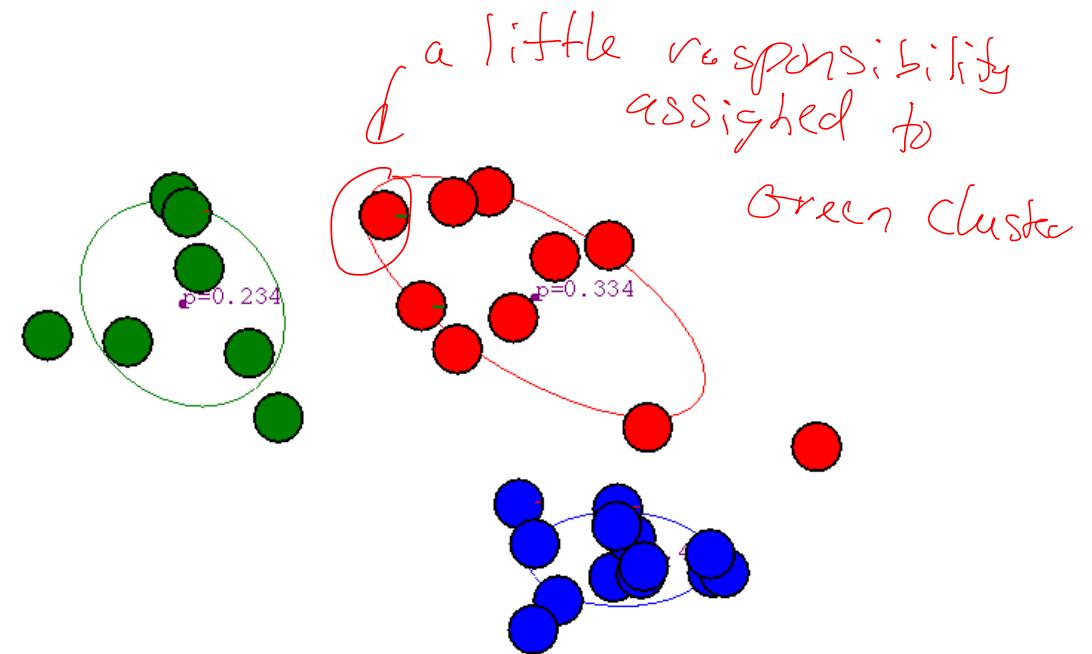
After 5th iteration



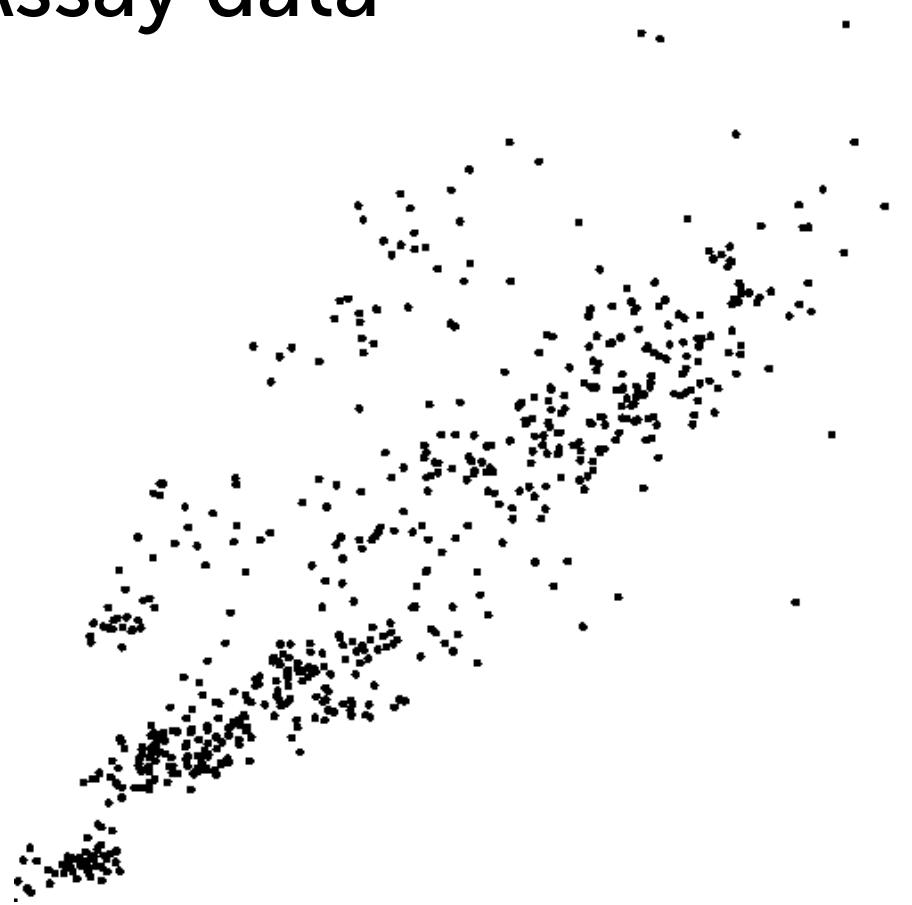
After 6th iteration



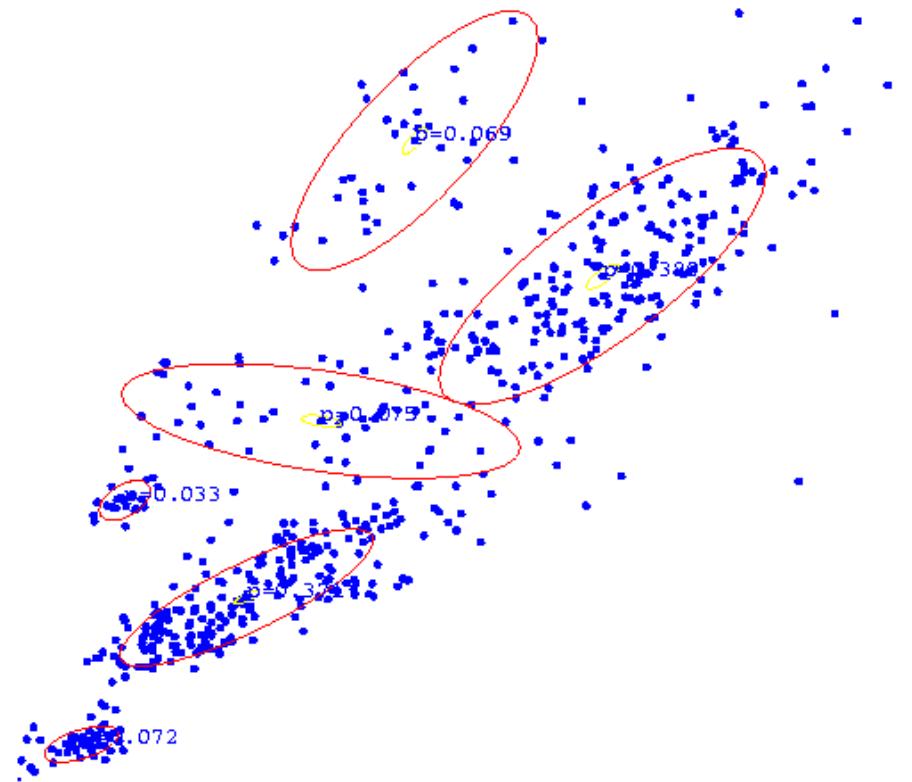
After 20th iteration



Some Bio Assay data



GMM clustering of the assay data



Resulting Density Estimator

